Atty. Dkt. No. 004-5639
1st Inventor: Laurent P. Daynes
Attorney: David W. O'Brien, (512) 347-9030

104111"1226560

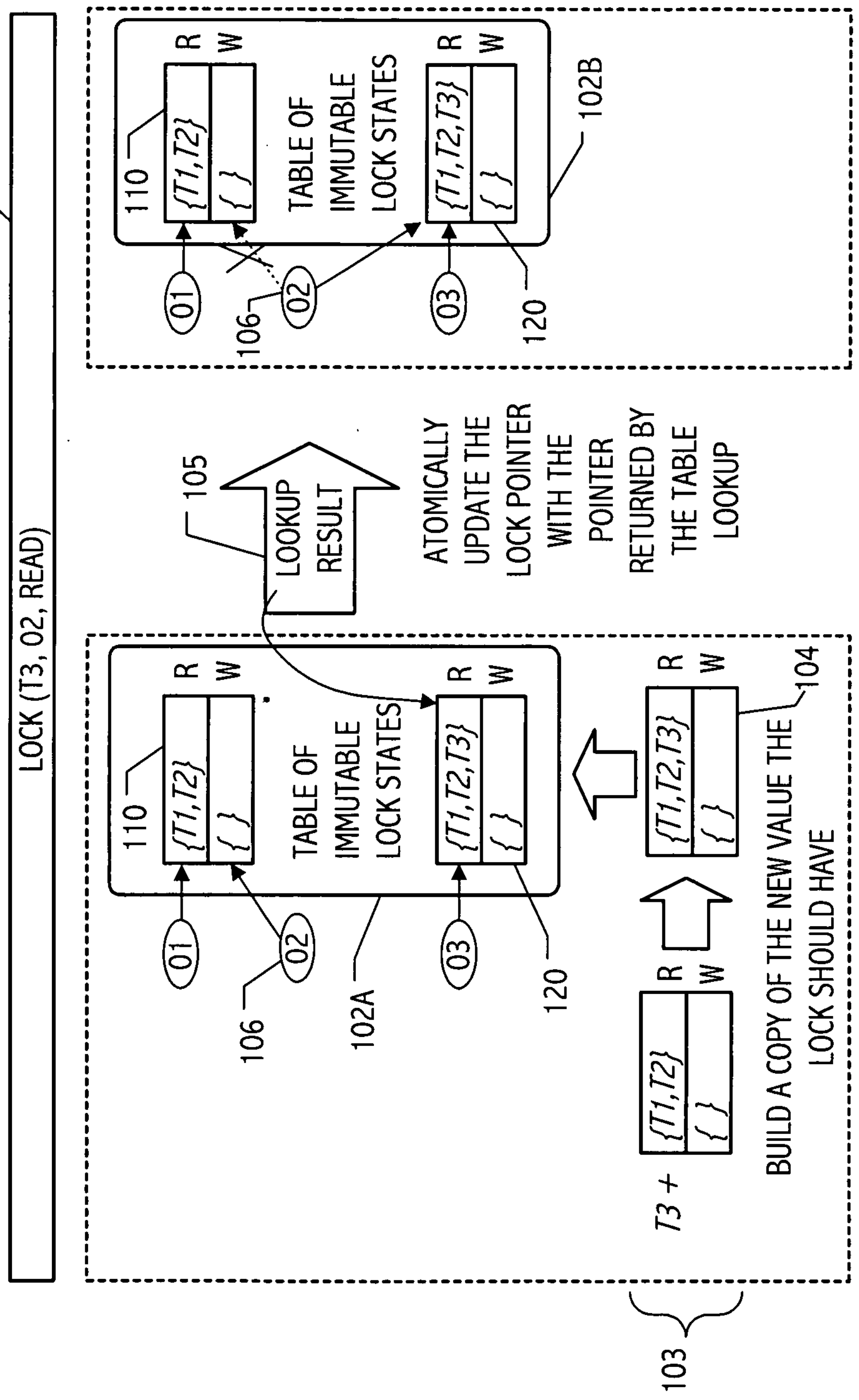
LOCK (T3, O2, READ)
101
103
O1
O2
O3
106
102A
110
120
{T1,T2}
{ }
{T1,T2,T3}
R
W
TABLE OF IMMUTABLE LOCK STATES
T3 +
104
BUILD A COPY OF THE NEW VALUE THE LOCK SHOULD HAVE
105
LOOKUP RESULT
ATOMICALLY UPDATE THE LOCK POINTER WITH THE POINTER RETURNED BY THE TABLE LOOKUP
102B

FIG. 1

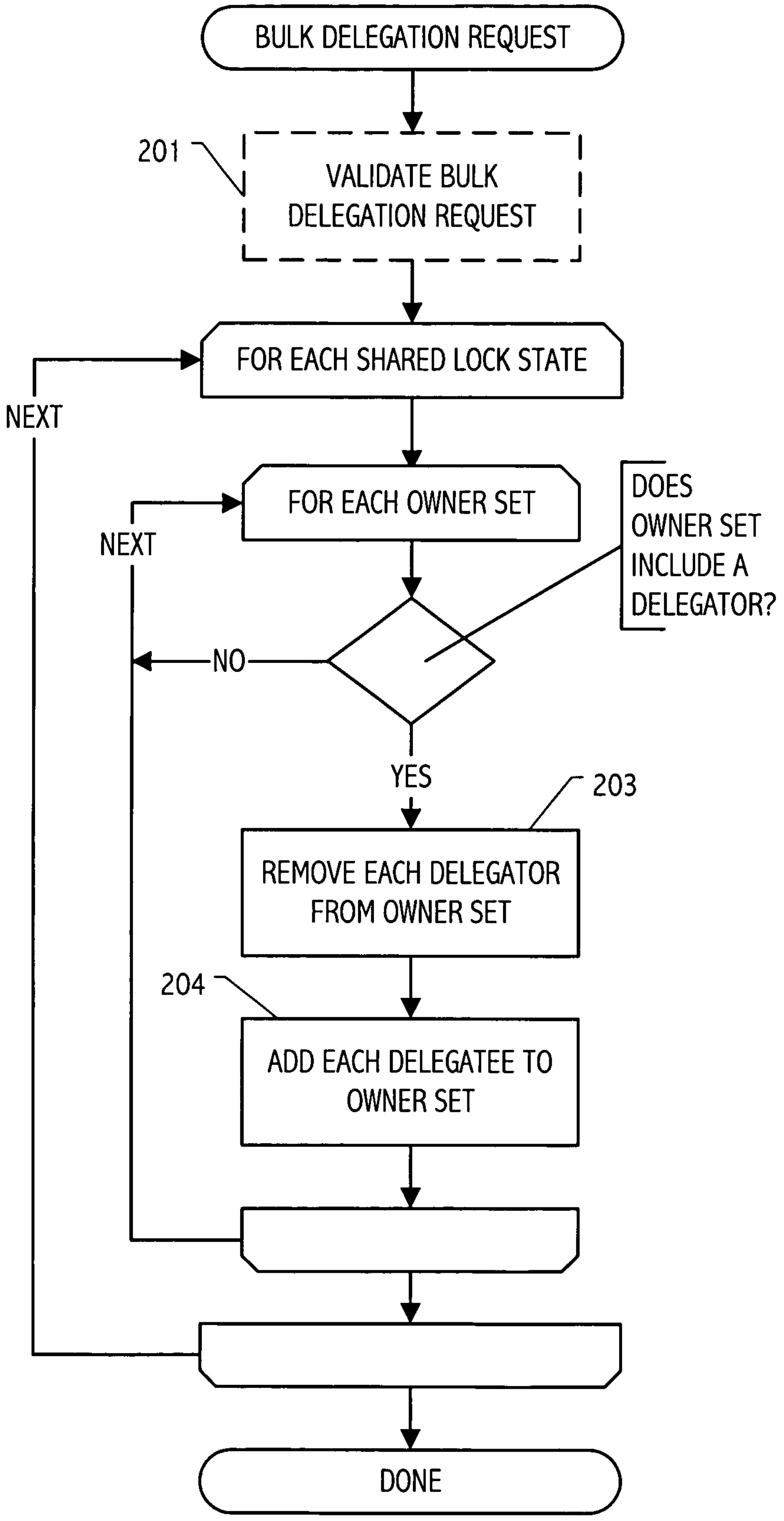
BULK DELEGATION REQUEST
201
VALIDATE BULK DELEGATION REQUEST
FOR EACH SHARED LOCK STATE
NEXT
FOR EACH OWNER SET
NEXT
DOES OWNER SET INCLUDE A DELEGATOR?
NO
YES
203
REMOVE EACH DELEGATOR FROM OWNER SET
204
ADD EACH DELEGATEE TO OWNER SET
DONE

FIG. 2

```
delegate(delegators, delegatees)
begin
  foreach l in TSLS
    if $\exists M$, $(delegators \cap Owners(l,M) \neq \emptyset) \wedge (\nexists M_i, M_i > M \wedge (delegators \cap Owners(l,M_i) \neq \emptyset))$
      TSLS.remove(l)
      // modify its owner set to reflect the effect of delegation
      foreach $M_i$, $M_i \leq M$
301      $Owners(l,M_i) \leftarrow [Owners(l,M_i) - delegators] \cup delegatees$
      end
      // does the new value duplicate an existing shared lock state ?
      if TSLS.contains(l)
        // yes. record the "original" being duplicated
        // and add the shared lock state to the set of duplicates.
        $original(l) \leftarrow$ TSLS.get(l)
        duplicates.add(l)
      else
        // no. Re-enter the modified shared lock in the TSLS.
        TSLS.add(l)
      endif
    endif
  end
  // Process duplicates now.
  foreach l in duplicates
    if $\exists M$, $(delegators \cap Owners(l,M) \neq \emptyset) \wedge (\nexists M_i, M_i > M \wedge (delegators \cap Owners(l,M_i) \neq \emptyset))$
      // modify its owner set to reflect the effect of delegation
      foreach $M_i$, $M_i \leq M$
        $Owners(l,M_i) \leftarrow [Owners(l,M_i) - delegators] \cup delegatees$
      end
    endif
  end
end
```

FIG. 3

```
// Determine the validity of a delegating a lock set to the value $l$
boolean isValid($delegators$, $delegatees$, $l$)
begin
  if $Owners(l, Write) = \emptyset$
    return true
  endif
  // at least one write lock owner
  if $Owners(l, Write) \cap delegators = \emptyset$
    // All the delegators are read owners.
    // The delegation is valid if all delegatees can ignore read-write
    // conflicts with the write owners.
    return $\forall t \in delegatees, Owners(l, Write) \subseteq ICW(t, rw)$
  endif
  // the lock is delegated in write mode – all delegatees must ignore
  // write-write conflicts between each others and with each remaining
  // owners of the lock in write mode. Also, write-read conflicts should
  // be ignored with remaining owners of the lock in read mode.
  if $|\, delegatees \,| > 1$
    // More than one delegatee
    if $\exists\ t \in delegatees,\ \exists\ c \in \{rw, wr, ww\},\ delegatees \not\subseteq ICW(t, c)$
      return false
    endif
  endif
  if $\exists\ t \in delegatees, (Owners(l, Write) - delegators) \not\subseteq ICW(t, ww)$
    return false
  endif
  if $\exists\ t \in delegatees, (Owners(l, Read) - delegators) \not\subseteq ICW(t, wr)$
    return false
  endif
  return true
end
```

FIG. 4

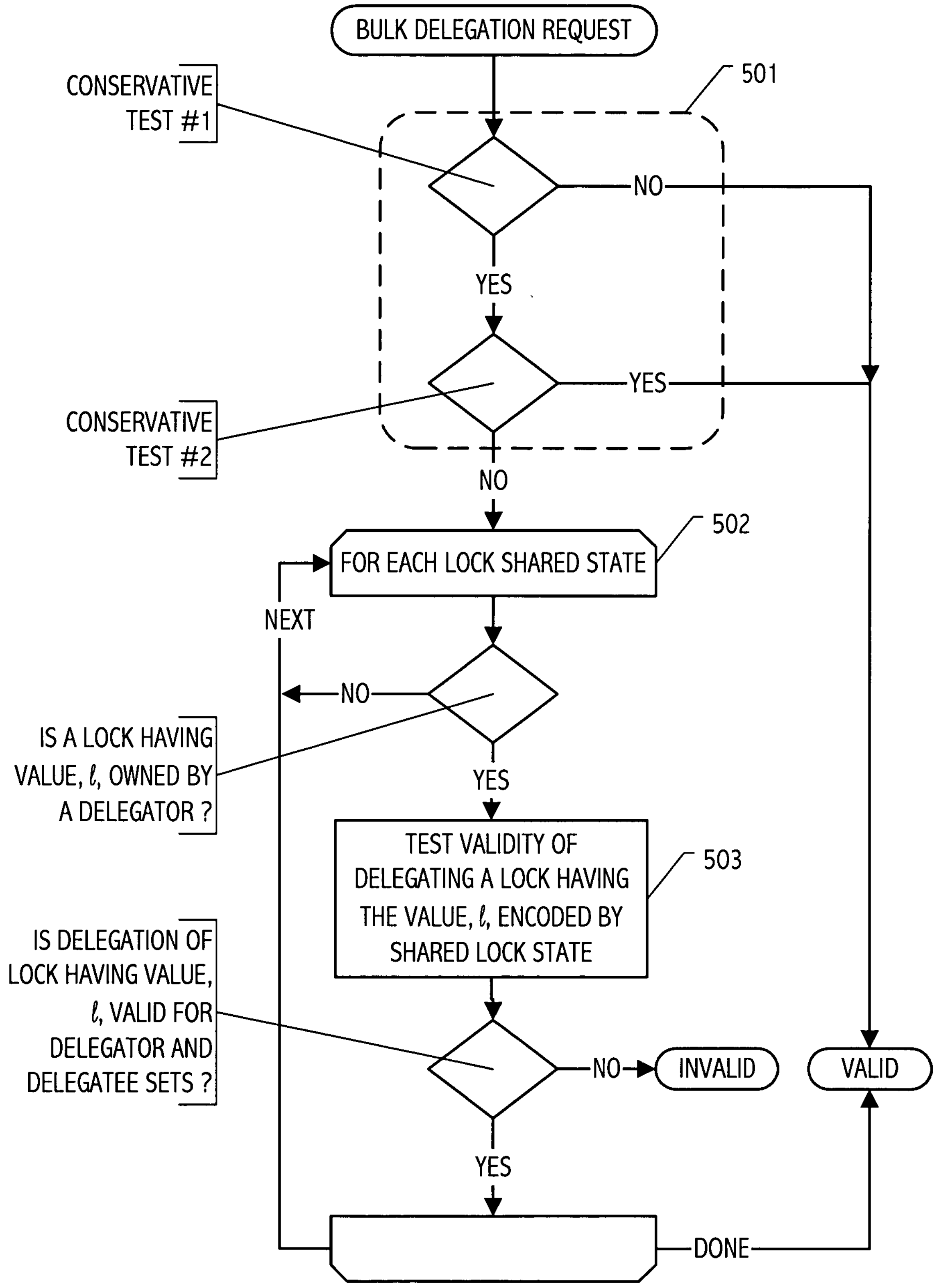
BULK DELEGATION REQUEST
501
CONSERVATIVE TEST #1
NO
YES
CONSERVATIVE TEST #2
YES
NO
502
FOR EACH LOCK SHARED STATE
NEXT
NO
IS A LOCK HAVING VALUE, ℓ, OWNED BY A DELEGATOR ?
YES
TEST VALIDITY OF DELEGATING A LOCK HAVING THE VALUE, ℓ, ENCODED BY SHARED LOCK STATE
503
IS DELEGATION OF LOCK HAVING VALUE, ℓ, VALID FOR DELEGATOR AND DELEGATEE SETS ?
NO
INVALID
VALID
YES
DONE

FIG. 5

```
     // Determine the validity of a bulk lock delegation
     boolean isValid($delegators$, $delegatees$)
     begin
602    if $wset \cap delegators = \emptyset$
         return true
603    else if ( $\forall\, t_d \in delegatees, \forall\, t_s \in delegators, \forall\, C \in \{rw, wr, ww\}$,
               $(ICW(t_s, C) - (delegators \cup \{t_d\})) \subseteq ICW(t_d, C)$
         // if at least one of the delegated lock is a write lock, the request
         // is valid only if the delegatees can ignore all conflicts which each other
         if $\exists\, l,\ Owners(l, Write) \cap delegators \neq \emptyset$
            return $(\forall\, t \in delegatees,\ \forall\, c \in \{rw, wr, ww\},\ delegatees \subseteq ICW(t, c))$
         else
            return true
         endif
       else
         // The two conservative tests have failed
         foreach $l$ in TSLS
            if $Owners(l, W) \neq \emptyset \wedge (\exists M,\ Owners(l, M) \cap delegators \neq \emptyset)$
601           if $\neg isValid(delegators, delegatees, l)$
                 return false
              endif
            endif
         end
       endif
       return true
     end
```

FIG. 6